Importance sampling strategy for non-convex randomized block-coordinate descent

Rémi Flamary, Alain Rakotomamonjy, Gilles Gasso

Importance Sampling Strategy for Non-Convex Randomized Block-Coordinate Descent

Rémi Flamary
Lagrange, UMR CNRS 7293, OCA
Université Côte d'Azur
France
remi.flamary@unice.fr

Alain Rakotomamonjy
LITIS Rouen/LIF Marseille
Université de Rouen - Université Aix-Marseille
France
alain.rakoto@insa-rouen.fr

Gilles Gasso
LITIS
INSA de Rouen
France
gilles.gasso@insa-rouen.fr

***Abstract*—As the number of samples and dimensionality of optimization problems related to statistics an machine learning explode, block coordinate descent algorithms have gained popularity since they reduce the original problem to several smaller ones. Coordinates to be optimized are usually selected randomly according to a given probability distribution. We introduce an importance sampling strategy that helps randomized coordinate descent algorithms to focus on blocks that are still far from convergence. The framework applies to problems composed of the sum of two possibly non-convex terms, one being separable and non-smooth. We have compared our algorithm to a full gradient proximal approach as well as to a randomized block coordinate algorithm that considers uniform sampling and cyclic block coordinate descent. Experimental evidences show the clear benefit of using an importance sampling strategy.**

I. Introduction

In the era of Big Data, current computational methods for statistics and machine learning are challenged by size of data both in terms of dimensionality and number of examples. Parameters of estimators learned from these large amount of data are usually obtained as minimizer of a regularized empirical risk problems of the form

$$\min_{\mathbf{x}\in\mathbb{R}^d} \{F(\mathbf{x}) = f(\mathbf{x}) + \lambda h(\mathbf{x})\} \tag{1}$$

where f is usually a smooth and non-convex function with Lipschitz gradient and h a non-smooth function. In such a large-scale and high-dimensionality context, most prevalent approaches use first-order method based on gradient descent [1] although second-order quasi-Newton algorithms have been considered [14].

More efficient algorithms can be considered for solving problem (1) if f and h present some special structures. When h is separable, Problem 1 can be expressed as

$$h(\mathbf{x}) = \sum_{i=1}^{m} h_i(\mathbf{x}_i)$$

We suppose that $\mathbf{x} \in \mathbb{R}^d$ is of the form $\mathbf{x} = [\mathbf{x}_1^\top, \dots \mathbf{x}_m^\top]^\top$ where m is the number of groups in $\mathbf{x}$ and $\mathbf{x}_i \in \mathbb{R}^{d_i}$ and $\sum_i d_i = d$. In this case, methods that can use the group structure such as coordinate descent algorithms [19] or randomized coordinate descent [12] are among the most efficient ones for solving problem (1).

In this paper, we focus on a specific class of randomized block proximal gradient algorithm, useful when each block h_i has a special structure. We suppose that each h_i is a difference of convex functions and is non-smooth. However, it has to have a closed-form proximal operator [8]. Such a situation mainly arises when $h(\mathbf{x})$ is a non-convex sparsity-inducing regularizer. Common non-convex and non-differentiable regularizers are the SCAD regularizer [6], the ℓ_p regularizer [9], the capped-ℓ_1 and the *log* penalty [4]. These regularizers have been frequently used for feature selection or for obtaining sparse models in machine learning [4], [7], [10].

A large majority of works dealing with randomized block coordinate descent algorithms (RBCD) considers uniform distribution of sampling [12], [15], [17]. Few attentions have been devoted to the use of arbitrary distribution [11], [13]. In these two latter efforts, principal statement is that the probability of drawing any block should be not less than a $p_{min} > 0$ value to ensure that all blocks have non-zero probabilities to be selected and hence to guarantee convergence in expectation of the algorithm. However, because no prior knowledge are usually available for directing the choice of the probability distribution of block sampling, experimental analysis of the randomized algorithms usually consider uniform distribution.

This paper proposes a probability distribution for randomized block coordinate sampling that goes beyond the uniform sampling and that is updated after each iteration of the algorithm. Indeed, we have designed a distribution that is dependent on approximate optimality condition of the problem. Owing to such a distribution, described in Section II we can bias the sampling towards coordinates that are still far from optimality allowing to save substantial computational efforts as illustrated by our empirical experiments (see Section III).

II. Framework and algorithm

A. Randomized BCD

We discuss now a generic approach for solving problem (1) when $h(\cdot)$ is separable by taking advantage of this separability. The general framework is shown in Algorithm 1 where $\nabla_i f(\mathbf{x})$ is the partial gradient at $\mathbf{x}$ of f with respect to $\mathbf{x}_i$.

At each iteration in the algorithm a block i is selected to be optimized (line 3). Then, a partial proximal gradient step is

Algorithm 1 Randomized Block Coordinate Descent (RBCD)

1: Set initial $\mathbf{x}^0$, $\theta > 0$, $\eta > 1$, $\sigma > 0$
2: **for** $k = 1, 2, \ldots$ **do**
3: $i \leftarrow$ randomly select current block from $\{1, 2, \ldots, m\}$ according to a probability distribution $\mathbf{p}$
4: $\mathbf{d} \leftarrow \mathbf{0}; \mathbf{d}_i \leftarrow \nabla_i f(\mathbf{x})$
5: $\mathbf{x}^k \leftarrow \text{prox}_{\frac{1}{\theta_k}h}(\mathbf{x}^{k-1} - \frac{1}{\theta_k}\mathbf{d})$,$j \leftarrow 0$
6: **while** $F(\mathbf{x}^k) > F(\mathbf{x}^{k-1}) - \frac{\sigma}{2}\|\mathbf{x}^k - \mathbf{x}^{k-1}\|$ **do**
7: $j \leftarrow j + 1$ and set $\gamma = (\eta)^j$
8: $\mathbf{x}^k \leftarrow \text{prox}_{\frac{1}{\theta^k \gamma}h}(\mathbf{x}^{k-1} - \frac{1}{\theta^k \gamma}\mathbf{d})$
9: **end while**
10: **end for**

Table I
FLOATING OPERATION AT EACH ITERATION FOR THE GIST AND RBCD FOR A LINEAR MODEL OF THE FORM $f(\mathbf{x}) = L(\mathbf{A}\mathbf{x})$. d_i IS THE DIMENSIONALITY OF THE GROUP i UPDATED AT THE CURRENT ITERATION.

Task	GIST	RBCD
Gradient computation	$2nd + n$	$2nd_i + n$
Proximal operator	d	d_i
Cost computation	$nd + n$	$nd_i + n$

performed (line 5) for the selected group. It consists in solving efficiently the proximal operator

$$\text{prox}_{\frac{1}{\theta}h}(\mathbf{v}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2 + \frac{1}{\theta}h(\mathbf{x}).$$

Note that since h is separable, the proximal operator can be applied only on the current group i and will update only $\mathbf{x}_i$. A backtracking (line 6-9) may be necessary to ensure a decrease in the objective F but a non monotone version can also be used as discussed in [11]. Finally, if the number of groups is set to 1, then the algorithm boils down to GIST [8], *i.e.* a proximal method for non-convex optimization.

This randomized algorithm is interesting *w.r.t.* the classical proximal gradient descent since it does not require the computation of the full gradient at each iteration. For instance, when estimating a linear model, the loss f can be expressed as $f(\mathbf{x}) = L(\mathbf{A}\mathbf{x})$. The gradient is $\nabla f(\mathbf{x}) = \mathbf{A}^\top L'(\mathbf{A}\mathbf{x})$ where the derivative L' is computed pointwise. Computing the partial gradient $\nabla_i f(\mathbf{x}) = \mathbf{A}_i^\top L'(\mathbf{A}\mathbf{x})$ where $\mathbf{A}_i$ is the submatrix of $\mathbf{A}$ corresponding to group i requires much less floating operations as reported in Table I since $d_i \ll d$. In addition, this computational complexity can be greatly decreased by storing the prediction $\mathbf{A}\mathbf{x}$ and by using the low complexity update $\mathbf{A}_i(\mathbf{x}_i^k - \mathbf{x}_i^{k-1})$ at each iteration.

B. Block selection and importance sampling

The convergence of the RBCD algorithm is clearly dependent of the block selection strategy of line 3 in Algorithm 1. One can select the group using classic cyclic rule as in [5], [3] or using the realization of a random distribution [12], [18]. The uniform distribution is often used in order to ensure that all blocks are updated equally, but convergence in expected value has been proved for any discrete distribution that have non-null components, ($p_{min} > 0$) [11].

In this work, we introduce a novel probability distribution for sampling blocks in RBCD. This distribution is dependent on the optimality conditions of each block. In other words, we want to update more often blocks that are still far from convergence. Formally, let $\mathbf{p} \in \mathbb{R}^{+m}$ be the discrete density distribution such that p_i is the probability that the block i is selected at a given iteration and $\sum_i p_i = 1$. We propose in this work to use the following distribution

$$p_i = \frac{\epsilon + (1 - \epsilon)\frac{z_i}{\|\mathbf{z}\|_\infty}}{m\epsilon + (1 - \epsilon)\frac{1}{\|\mathbf{z}\|_\infty}\sum_i z_i} \tag{2}$$

where $\epsilon \in (0, 1]$ is a user-defined parameter, $\mathbf{z} \geq \mathbf{0}$ is a vector composed of coordinates $\{z_i\}_{i=1}^m$ and $\|\mathbf{z}\|_\infty = \max_i |z_i|$ is the infinite norm. As made clearer in the sequel, a component z_i encodes the optimality condition violation in each block. Indeed, let $h_i = h_{i,1} - h_{i,2}$, with $h_{i,1}$ and $h_{i,2}$ being two convex functions, then if $\mathbf{x}^\star$ is a local minimizer of $F(\mathbf{x})$, from Clarke subdifferential calculus [16], one can show that a necessary condition of optimality is that there exists $\mathbf{v} \in \partial h_{i,1}(\mathbf{x}^\star)$ and $\mathbf{u} \in \partial h_{i,2}(\mathbf{x}^\star)$ such that $0 \in \nabla_i f(\mathbf{x}^\star) + \lambda\mathbf{v} - \lambda\mathbf{u}$ for all i. Accordingly, we define the optimality condition violation z_i as

$$z_i = \min_{\mathbf{v} \in \partial h_{i,1}(\mathbf{x}), \mathbf{u} \in \partial h_{i,2}(\mathbf{x})} \|\nabla_i f(\mathbf{x}) + \lambda\mathbf{v} - \lambda\mathbf{u}\|_\infty \tag{3}$$

The role of ϵ in Equation (2) is to balance the effect of the optimality condition on the distribution. When $\epsilon = 1$, we retrieve a uniform distribution. Other values of ϵ will ensure that if a variable in a block has not converged, its block is likely to be updated more often than a block that has converged. Note that, owing to the DC decomposition of h, the violation (3) can be easily computed, even for non-convex penalty function such as SCAD or the log-sum as discussed in [2].

Computing the optimality condition violation vector $\mathbf{z}$ is not possible in practice for RBCD since it requires the full gradient of the problem, which as discussed in the previous section, is not computed at each iteration. As a solution, we propose to use a vector $\tilde{\mathbf{z}}$ initialized with the exact condition violation computed from the initial vector $\mathbf{x}^0$. Thereon, only the i^{th} entry of $\tilde{\mathbf{z}}$ is updated at each iteration leading to an approximate optimality condition evaluation. Indeed, in algorithm 1 line 4, when a partial gradient $\nabla_i f(\mathbf{x})$ is computed, we can use it to update the approximate $\tilde{z}_i$ and then update the probabilities $\mathbf{p}$ accordingly. This latter vector is clearly a coarse approximation of the optimality condition violation but as shown in the experiments it is a relevant choice for the proposed importance sampling scheme.

C. On tricks of the trade

The proposed optimization algorithm has an important parameter that has to be chosen carefully: the initial gradient step size $1/\theta^k$ at each iteration. If chosen too small, the gradients steps will barely improve the objective value, if chosen too large the backtracking step in lines 6-9 will require numerous computation of the loss function. In this work we

use an extension of the Barzilai-Borwein (BB) rule that has been proposed in a non-convex scheme by [8]. This approach consists in using a Newton step with the approximate Hessian $\sigma\mathbf{I}$. When performing the full gradient descent in GIST, the BB rule gives

$$\theta^{k+1} = \frac{\Delta\mathbf{x}^\top \Delta\mathbf{g}}{\Delta\mathbf{x}^\top \Delta\mathbf{x}} \tag{4}$$

where $\Delta\mathbf{x} = \mathbf{x}^k - \mathbf{x}^{k-1}$ and $\Delta\mathbf{g} = \nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^{k-1})$. Again, in our algorithm the full gradient is not available but we can still benefit from the second-order approximation brought to us by the BB rule. We propose to this end to model the Hessian as a diagonal matrix where the weight of the diagonal is block-dependent. In other word, we store an estimate $\boldsymbol{\theta} \in \mathbb{R}^{+m}$ whose components θ_i are updated similarly to equation (4) but using instead partial gradient and variations $\Delta\mathbf{x}_i = \mathbf{x}_i^k - \mathbf{x}_i^{k-1}$ and $\Delta\mathbf{g}_i = \nabla_i f(\mathbf{x}^k) - \nabla_i f(\mathbf{x}^{k-1})$. This new rule is actually more general than the classical BB-rule since it brings local information and encodes a more precise Hessian approximation with group-wise coefficients similar to the variable metric in [5] .

III. Numerical Experiments

In this section, we illustrate the behaviour of our randomized BCD algorithm with importance sampling on some toy and real-world classification problems. For all problems, we have considered a logistic loss function and the log-sum non-convex sparsity inducing penalty defined as

$$h(\mathbf{x}) = \rho \sum_i^d \log\left(1 + \frac{|x_i|}{\rho}\right)$$

with $\rho > 0$. We have compared our algorithm to a non-convex proximal gradient algorithm known as GIST [8] and a randomized BCD version of GIST with uniform sampling [11]. Note that since this regularization term is fully separable per variable, we used a separation of m blocks of size $\frac{d}{m}$ variables.

A. Toy problem

As in [14] we consider a binary classification problem in $\mathbb{R}^d$. Among these d variables, only T of them define a subspace of $\mathbb{R}^d$ in which classes can be discriminated. For these T relevant variables, the two classes follow a Gaussian pdf with means respectively $\boldsymbol{\mu}$ and $-\boldsymbol{\mu}$ and covariance matrices randomly drawn from a Wishart distribution $W(\mathbf{I}, T)$ where $\mathbf{I}$ is the identity matrix. The components of $\boldsymbol{\mu}$ have been independently and identically drawn from $\{-1, +1\}$. The other $d - T$ non-relevant variables follow an i.i.d Gaussian probability distribution with zero mean and unit variance for both classes. We have respectively sampled n and n_t = 1000 number of examples for training and testing. Before learning, the training set has been normalized to zero mean and unit variance and test set has been rescaled accordingly. Note that the hyperparameter λ or any other parameters related to the regularization term have been set so as to maximize the performance of the GIST algorithm on the test set. We have initialized all algorithms with the zero vector ($\mathbf{x}^0 = \mathbf{0}$).

The different algorithms have been compared based on their computational demands and more exactly based on the number of flops they need for reaching a stopping criterion. Hence, this criterion is critical for a fair comparison. The GIST algorithm has been run until it reaches a necessary optimality condition $\|\mathbf{z}\|_\infty$ lower than 10^{-3} or until 1000 iterations is attained. For the randomized algorithms, including our approach , the stopping criterion is set according to a maximal number of iterations. This number is set so that the number of coordinate gradient evaluations is equal for all algorithms i.e we have used the number of GIST iterations$\times m$ where m is the number of blocks. In the sequel, the number of flops reported is related to those needed for computing both function values and gradient evaluations.

Figure 1 (left) presents some examples of optimality condition $\|\mathbf{z}\|_\infty$ evolution with respects to the number of flops. These curves are obtained as averages over 20 iterations of the results obtained for a given experimental set-up (here $n = 200$, $d = 2000$ and $T = 20$). We can first note that with respect to optimality condition, RBCD algorithm with uniform sampling (Unif RBCD) behaves similarly to the GIST algorithm and a cyclic BCD (Cyclic BCD). In terms of flops, few gain can be expected from such an approach. Instead, using importance sampling (IS RBCD) considerably helps in improving convergence. Such a behaviour can also be noted when monitoring evolution of the objective value (see central panel in Figure 1). Randomized algorithms tend to converge faster towards their optimal value with a clear advantage to the importance sampling approach. Finally, while they are not reported due to lack of space, the final classification performances are similar for all three methods.

Figure 1 (right) depicts evolutions of optimality conditions depending on block-coordinate group size. We can note that regardless of this size, our importance sampling approach achieves better performance than the GIST algorithm. In addition, it is clear that for our examples, the smaller the size is, the faster convergence we obtain.

B. Real-world classification problems

We have also compared these algorithms on real-world high-dimensional learning problems. The related datasets have been already used as benchmark datasets in [8], [14]. For these problems, we have used 80% of the examples as training set and the remaining as test set. Again, hyperparameters of the model have been chosen so as to roughly maximize performances of the GIST algorithm. Stopping criteria of all algorithms have been set as previously. However, maximal number of iterations has been set to 5000 for GIST. In addition, we have limited the maximal number of iterations to 20000. The number of blocks has been set to $m = 100$ for all datasets.

Performances of the different algorithms are reported in Table II. Three measure of performances have been compared. Classification rates of all algorithms are almost similar although differences in performances are statistically significant in favor of GIST according to a Wilcoxon sign rank test with a p-value of 0.05. We explain this by the fact that regulariza-

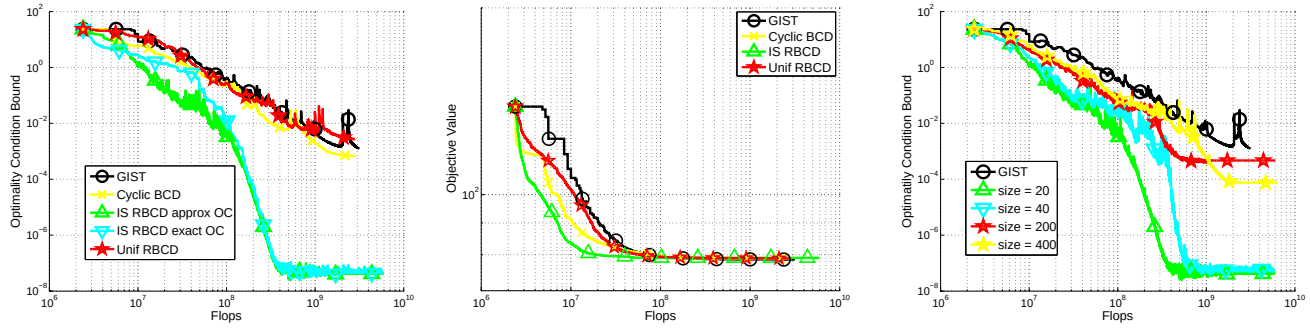

Figure 1. Example of (left) optimality condition violation and (middle) objective value evolution with respects to the number of flops, averaged over 20 iterations and with blocks of size 20. For the left panel, we have plotted the exact violation (computed with $\mathbf{z}$) as well as the approximated one (computed with $\tilde{\mathbf{z}}$). (right) Optimality conditions violation averaged over 20 iterations for different block sizes used in IS RBCD. (Best viewed in color)

Table II

COMPARISON OF GIST AND RANDOMIZED BCD ALGORITHMS ON REAL-WORLD BENCHMARK PROBLEMS. THE FIRST COLUMNS OF THE TABLE PROVIDE THE NAME OF THE DATASETS, THE NUMBER OF TRAINING EXAMPLES n AND THEIR DIMENSIONALITY d. THREE MEASURES OF PERFORMANCES ARE PROVIDED : THE CLASSIFICATION RATE, THE NUMBER OF FLOPS NEEDED FOR CONVERGENCE, THE OPTIMALITY CONDITION. THE OBJECTIVE VALUE IS GIVEN FOR A SAKE OF INFORMATION BUT IT IS NOT A RELEVANT CRITERION IN A NON-CONVEX PROBLEM.

data	n	d	Algorithm	Class. Rate (%)	Flops $\times 10^9$	Opt. Condition	Obj. Val
classic	7094	41681	GIST	96.37±0.5	9277.76±64.6	0.03±0.0	32.64±2.2
classic	7094	41681	IS RBCD	95.11±0.7	347.16±4.1	0.01±0.0	25.23±0.8
classic	7094	41681	Unif RBCD	95.87±0.6	364.12±66.2	0.03±0.0	35.26±0.8
la2	3075	31472	GIST	91.11±1.1	3148.75±287.8	0.06±0.1	39.42±57.7
la2	3075	31472	IS RBCD	90.98±1.2	101.16±3.6	0.15±0.2	43.35±59.0
la2	3075	31472	Unif RBCD	91.04±0.9	108.11±4.8	0.23±0.3	45.51±59.0
ohscal	11162	11465	GIST	88.30±0.6	7452.22±895.6	2.65±2.3	520.41±451.2
ohscal	11162	11465	IS RBCD	87.88±0.8	164.42±21.5	0.87±0.6	480.53±428.5
ohscal	11162	11465	Unif RBCD	87.75±0.8	156.45±17.7	1.14±1.1	480.55±428.5
sports	8580	14870	GIST	97.93±0.4	5034.75±1219.5	0.11±0.1	208.11±215.2
sports	8580	14870	IS RBCD	97.76±0.5	154.74±20.3	0.07±0.1	212.05±215.3
sports	8580	14870	Unif RBCD	97.86±0.4	173.99±10.6	0.39±0.3	222.38±215.3

tion parameters have been selected *w.r.t.* to its generalization performances. The number of flops needed for convergence are highly in favor of the randomized algorithms. The factor gain in flops ranges in between 26 to 45. Interestingly, exact optimality conditions after algorithms have halted are always in favor of our importance sampling randomized BCD algorithms except for the *la2* dataset. Note that, in the table, we have also provided the objective values of the algorithms upon convergence. As one may have expected in a non-convex optimization problem, different "nearly" optimal objective values leads to similar classification rate performances stressing the existence of several local minimizers with good generalization property.

IV. CONCLUSION

This paper introduced a framework for randomized block coordinate descent algorithm that leverages on importance sampling. We presented a sampling distribution that biases the algorithm to focus on block coordinates that are still far from convergence. While this idea is rather simple, our experimental results have shown that it considerably helps in achieving a faster empirical convergence of the randomized BCD algorithm. Future works will be devoted to the theoretical analysis of the importance sampling impact on the convergence rate. In addition, we plan to carry out thorough experimental analyses that unveil the impact of the algorithm parameters.

REFERENCES

[1] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[2] A. Boisbunon, R. Flamary, and A. Rakotomamonjy, "Active set strategy for high-dimensional non-convex sparse optimization problems," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*. IEEE, Firenze, Italy, May 2014, pp. 1517–1521.

[3] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[4] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

[5] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function," *Journal of Optimization Theory and Applications*, vol. 162, no. 1, pp. 107–132, 2014.

[6] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[7] G. Gasso, A. Rakotomamonjy, and S. Canu, "Recovering sparse signals with a certain family of non-convex penalties and dc programming," *IEEE Trans. Signal Processing*, vol. 57, no. 12, pp. 4686–4698, 2009.

[8] P. Gong, C. Zhang, Z. Lu, J. Huang, and Y. Jieping, "A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, Jun. 2013, pp. 37–45.

[9] K. Knight and W. Fu, "Asymptotics for lasso-type estimators," *Annals of Statistics*, vol. 28, no. 5, pp. 1356–1378, 2000.

[10] L. Laporte, R. Flamary, S. Canu, S. Dejean, and J. Mothe, "Nonconvex regularizations for feature selection in ranking with sparse svm," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 6, pp. 1118–1130, 2014.

[11] Z. Lu and L. Xiao, "A randomized nonmonotone block proximal gradient method for a class of structured non linear programming," *Arxiv*, no. 1306.5918v2, 2015.

[12] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[13] Z. Qu and P. Richtárik, "Coordinate descent with arbitrary sampling i: Algorithms and complexity," *arXiv preprint arXiv:1412.8060*, 2014.

[14] A. Rakotomamonjy, R. Flamary, and G. Gasso, "Dc proximal newton for nonconvex optimization problems," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 1, no. 1, pp. 1–13, 2015.

[15] P. Richtárik and M. Takávc, "Efficient serial and parallel coordinate descent methods for huge-scale truss topology design," in *Operations Research Proceedings 2011*. Springer, 2012, pp. 27–32.

[16] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.

[17] S. Shalev-Shwartz and A. Tewari, "Stochastic methods for l 1-regularized loss minimization," *The Journal of Machine Learning Research*, vol. 12, pp. 1865–1892, 2011.

[18] R. Tappenden, P. Richtárik, and J. Gondzio, "Inexact coordinate descent: Complexity and preconditioning," *ArXiv e-prints*, 2013.

[19] P. Tseng, "Convergence of block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Application*, vol. 109, pp. 475–494, 2001.